IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief-Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450, on February 5, 2007.

Robert E. Malm

In re Application of: CUREY et al. Serial Number: 09/821,537 Filing Date: 03/28/2001 For: PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS	Group Art Unit: 2152 Examiner: PHILIP C. LEE Telephone: (703) 305-8646

REPLY BRIEF TRANSMITTAL LETTER

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

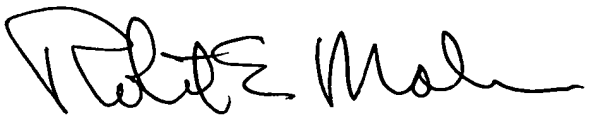
Dear Sir:

In response to the Examiner's Answer dated 12/13/06, appellants hereby submit a reply brief pursuant to 37 CFR § 41.41.

The Reply Brief is in compliance with the requirements set forth in 37 CFR § 41.41. The Reply Brief does not include any new or non-admitted amendment, or any new or non-admitted

affidavit or other evidence.

Respectfully submitted,



Robert E. Malm
Reg. No. 34,662

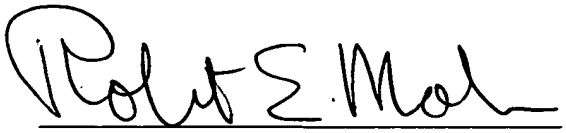
16624 Pequeno Place
Pacific Palisades, CA 90272
Tel. No: (310) 573-1781

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

CERTIFICATE OF MAILING

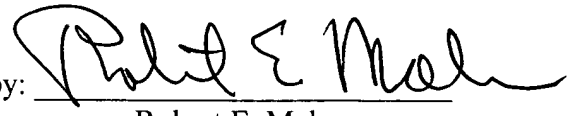
I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, Mail Stop Appeal Brief - Patents, Box 1450, Alexandria, VA 22313-1450 on February 5, 2007.

Robert E. Malm



In re Application of: RANDALL K. CUREY et al. Serial Number: 09/821,537 Filing Date: 03/28/2001 For: PARTITIONED EXECUTIVE STRUCTURE FOR REAL-TIME PROGRAMS	Group Art Unit: 2152 Examiner: PHILIP C. LEE Telephone: 703 308 7990

REPLY BRIEF

Submitted by: 

Robert E. Malm

Mailing Address: 16624 Pequeno Place
Pacific Palisades, CA 90272

Telephone: (310) 573-1781

STATUS OF CLAIMS

Claims 1-49 are pending in the application.

Claims 1-49 were rejected and all are subject to appeal.

Rejection of claims 2, 6-7, 27, and 31-32 withdrawn by examiner (see GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL for details).

GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

I. WHETHER CLAIMS 1, 25, AND 26 ARE UNPATENTABLE UNDER NON-STATUTORY DOUBLE-PATENTING DOCTRINE IN VIEW OF MARK ET AL. (U.S. 6,829,763).

Rejections withdrawn by examiner. Examiner's Answer, In reply to Argument (1)

II. WHETHER CLAIMS 1-49 ARE UNPATENTABLE UNDER 35 U.S.C. § 112, SECOND PARAGRAPH, AS BEING INDEFINITE.

Rejections of claims 2, 6, 7,18, 25, 27, 31, 32, 43 withdrawn by examiner. Examiner's Answer, In reply to Arguments (6), (8), (9), (11)

III. WHETHER CLAIM 25 IS UNPATENTABLE UNDER 35 U.S.C. § 112, SECOND PARAGRAPH, FOR FAILING TO SET FORTH THE SUBJECT MATTER WHICH APPLICANTS REGARD AS THEIR INVENTION.

IV. WHETHER CLAIM 25 IS UNPATENTABLE UNDER 35 U.S.C. § 101 BECAUSE IT EMBRACES OR OVERLAPS TWO DIFFERENT STATUTORY CLASSES.

Rejections withdrawn by examiner. Examiner's Answer, In reply to Argument (12)

V. WHETHER CLAIMS 1, 8, 23, 25, 26, 33, AND 48 ARE UNPATENTABLE UNDER 35 U.S.C. § 102(B) AS BEING ANTICIPATED BY BLUM ET AL. (U.S. 4,109,311).

VI. WHETHER CLAIMS 10-11, 13-14, 18, 35-36, 38-39, AND 43 ARE UNPATENTABLE UNDER 35 U.S.C. § 103(A) IN VIEW OF BLUM ET AL. (U.S. 4,109,311).

ARGUMENT

I. WHETHER CLAIMS 1, 25, AND 26 ARE UNPATENTABLE UNDER NON-STATUTORY DOUBLE-PATENTING DOCTRINE IN VIEW OF MARK ET AL. (U.S. 6,829,763).

Examiner's Reply to Argument 1, Answer, p. 11

The examiner withdrew the non-statutory double-patenting rejections of claims 1, 25, and 26.

II. WHETHER CLAIMS 1-49 ARE UNPATENTABLE UNDER 35 U.S.C. § 112, SECOND PARAGRAPH, AS BEING INDEFINITE.

CLAIMS 1 AND 26

Claims 1 and 26 read as follows:

1. A method for repetitively executing a plurality of software packages at one or more rates, utilizing a common set of computational resources, the method comprising the steps:

generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages;

executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals.

26. Apparatus for repetitively executing a plurality of software packages at a plurality of rates, the apparatus comprising:

a means for generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages;

a means for executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals.

Examiner's Reply to Argument 2, Answer, p. 11

The preamble states that the invention is "a method (or apparatus) for repetitively executing a plurality of software packages." To execute this plurality of software packages, each of this plurality

of software packages must be assigned a sequence of time intervals for its execution. This requirement is specified in the first element of both claims by using the term "the plurality of software packages" referring to "a plurality of software packages in the preamble.

However, after the user has arranged for the assignment of a sequence of time intervals to each software packages, he may decide to execute only some of these software packages. The software packages to be executed are referred to in the second elements of the claims as "a plurality of software packages" and are identifiable because they are those software packages from "the plurality of software packages" referred to in element 1 that are to be executed. The plurality of software packages to be executed is a subset of "the plurality of software packages" referred to in the first element of each claim. We know it is a subset because the software packages to be executed (second element) must each have been assigned a sequence of time intervals as a result of the first element.

CLAIMS 21 AND 46

Claims 21 and 46 read as follows:

21. The method of claim 1 wherein an executive software package enforces the discipline that each software package executes only during the time intervals of its sequence of time intervals, the executive software package determining when the execution of a software package extends into a time interval belonging to the sequence of time intervals assigned to another software package and performs a remedial action.

46. The apparatus of claim 26 wherein an executive software package enforces the discipline that each software package executes only during the time intervals of its sequence of time intervals, the executive software package determining when the execution of a software package extends into a time interval belonging to the sequence of time intervals assigned to another software package and performs a remedial action.

Examiner's Reply to Argument 3, Answer, p. 12.

In "that each software package executes", the term "each software package" refers obviously to a member of the plurality of software packages that executes as specified in element 2 of the claims.

In "when the execution of a software package extends", the term "a software package" obviously refers obviously to a member of the plurality of software packages that executes as specified in element 2 of the claims.

In "time intervals assigned to another software package", the term "another software package" refers obviously to a member of the plurality of software packages that has been assigned a sequence of time intervals for execution in element 1 of the claims.

CLAIMS 23 AND 48

Claims 23 and 48 read as follows:

23. The method of claim 1 wherein one or more of the plurality of software packages are independently compiled, linked, and loaded.

48. The apparatus of claim 26 wherein one or more of the plurality of software packages are independently compiled, linked, and loaded.

Examiner's Reply to Argument 4, Answer, pp. 12-13.

The examiner seems concerned that a software package that is to be executed (which is obviously a member of "a plurality of software packages" to be executed in element 2 of the claims) is also a member of "the plurality of software packages" to which sequences of time intervals have been assigned in element 1 of the claims. But the examiner should not be concerned. Software packages that are to be executed (those of element 2) must have assigned sequences of time intervals

and must therefore be members of "the plurality of software packages" specified in element 1 of the claims.

What IS true is that each software package that has been assigned a sequence of time intervals in element 1 will not necessarily be executed in element 2.

CLAIMS 24 AND 49

Claims 24 and 49 read as follows:

24. The method of claim 1 wherein a software package has its own stack, the software package's stack being selected prior to executing the software package.

49. The apparatus of claim 26 wherein a software package has its own stack, the software package's stack being selected prior to executing the software package.

Examiner's Reply to Argument 5, Answer, p. 13.

Please see applicants' response above to Argument 4.

CLAIMS 2 AND 27

Claims 2 and 27 read as follows:

2. The method of claim 1 wherein the plurality of software packages of the "executing" step includes only valid software packages, the method further comprising the step:
utilizing one or more tests to identify the software packages that are valid.

27. The apparatus of claim 26 wherein the plurality of software packages executed by the "executing" means includes only valid software packages, the apparatus further comprising:
a means for utilizing one or more tests to identify the software packages that are valid.

Examiner's Reply to Argument 6, Answer, p. 13.

The examiner withdrew the 35 U.S.C. 112, second paragraph, rejections of claims 2 and 27.

CLAIMS 3 AND 28

Claims 3 and 28 read as follows:

3. The method of claim 2 wherein one of the tests for validity is a one's complement checksum test of a software package's program memory.

28. The apparatus of claim 27 wherein one of the tests for validity is a one's complement checksum test of a software package's program memory.

Examiner's Reply to Argument 7, Answer, pp. 13-14.

The examiner objects to the description of the test as a "checksum test of a software package's program memory." He argues that it inclear how a checksum rocess can be applied on a memory of a computer system.

A "checksum" is a term of art in the mathematics of computing which means a sum obtained by adding the digits in a numeral, or group of numerals. A "checksum test" is performed by comparing this sum with a previously computed value to verify that no errors have occurred. From the definition alone, the answer to the examiner's question is obvious. By reading into computer memory data together with checksums and then reading out the data and checksums one can verify if errors have occurred in the read-in/read-out process.

Utilizing a checksum test of a software package's program memory is equivalent to utilizing a recall test of a person's memory. For example, if a person has had a blow to the head, one might ask the person what his name is. The results of this test is a test of the person's memory function just as the checksum test is a test of a software packages program memory function.

A person's memory peforms the same functions for a person as a software package's program memory does for a software package. In the case of a person, the information stored in a person's memory enables the person to perform the daily tasks associated with living. In the case of a

software package, the data stored in the software package's program memory enables the software package to perform the tasks that have been assigned to it.

There is nothing unclear about the language of claims 3 and 28.

CLAIMS 6-7 AND 31-32

Claims 6-7 and 31-32 read as follows:

6. The method of claim 2 wherein a software package is assigned its own dedicated memory region, the software package's dedicated memory region including a stack memory region and/or a heap memory region, one of the tests for validity being whether the stack memory range and/or the heap memory range assigned during the execution of the software package's initialization procedure and the various associated entry points lies within the software package's dedicated memory region.

7. The method of claim 6 wherein one of the tests is whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time.

31. The apparatus of claim 27 wherein a software package is assigned its own dedicated memory region, the software package's dedicated memory region including a stack memory region and/or a heap memory region, one of the tests for validity being whether the stack memory range and/or the heap memory range assigned during the execution of the software package's initialization procedure and the various associated entry points lies within the software package's dedicated memory region.

32. The apparatus of claim 31 wherein one of the tests is whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time.

Examiner's Reply to Argument 8, Answer, p. 14.

The examiner withdrew the 35 U.S.C. 112, second paragraph, rejections of claims 6-7 and 31-32.

CLAIMS 18 AND 43

Claims 18 and 43 read as follows:

18. The method of claim 1 wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software.

43. The apparatus of claim 26 wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software.

Examiner's Reply to Argument 9, Answer, p. 14.

The examiner withdrew the 35 U.S.C. 112, second paragraph, rejections of claims 18 and 43.

CLAIMS 22 AND 47

Claims 22 and 47 read as follows:

22. The method of claim 1 wherein the presence of those software packages that are present is detected.

47. The apparatus of claim 26 wherein the presence of those software packages that are present is detected.

Examiner's Reply to Argument 10, Answer, p. 14.

Applicants do indeed intend "those software packages" to refer to a subset of the plurality of software packages referred to in elements 2 in claims 1 and 26 as being the software packages intended for execution. In most cases, "those software packages" will be the complete set of software packages to be executed. However, humans make errors and one or more of the software packages intended for execution may not have been loaded into memory, in which case "those software packages that are present" will be a subset of the plurality of software packages intended for execution.

III. WHETHER CLAIM 25 IS UNPATENTABLE UNDER 35 U.S.C. § 112, SECOND PARAGRAPH, FOR FAILING TO SET FORTH THE SUBJECT MATTER WHICH APPLICANTS REGARD AS THEIR INVENTION.

Claim 25 reads as follows:

25. Apparatus for practicing the method of claim 1.

Examiner's Reply to Argument 11, Answer, p. 14.

The examiner withdrew the 35 U.S.C. 112, second paragraph, rejection of claim 25..

IV. WHETHER CLAIM 25 IS UNPATENTABLE UNDER 35 U.S.C. § 101 BECAUSE IT EMBRACES OR OVERLAPS TWO DIFFERENT STATUTORY CLASSES.

Claim 25 reads as follows:

25. Apparatus for practicing the method of claim 1.

Examiner's Reply to Argument 12, Answer, p. 14.

The examiner withdrew the 35 U.S.C. 101 rejection of claim 25.

V. WHETHER CLAIMS 1, 8, 23, 25, 26, 33, AND 48 ARE UNPATENTABLE UNDER 35 U.S.C. § 102(B) AS BEING ANTICIPATED BY BLUM ET AL. (U.S. 4,109,311).

"A claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." *Verdegaal Bros. v. UnionOil Co. of California,* 2 USPQ2d 1051, 1053 (Fed. Cir. 1987). (Cited in Manual of Patent Examining Procedure, 8th Edition, May 2004 Revision, § 2131.)

"The identical invention must be shown in as complete detail as is contained in the. . . claim." *Richardson v. Suzuki Motor Co.*, 9 USPQ2d 1913, 1920 (Fed. Cir. 1989). (Cited in Manual of Patent Examining Procedure, 8th Edition, May 2004 Revision, § 2131.)

CLAIMS 1, 25, AND 26

Claim 1 reads as follows:

*1. A method **[1] for repetitively executing a plurality of software packages at one or more rates**, utilizing a common set of computational resources, the method comprising the steps:*

[2] generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages;

[3] executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals.

Apparatus claims 25 and 26 also contain limitations [1], [2], and [3].

The limitations shown in boldface are not disclosed by Blum et al.

Limitation [1]

Blum et al. does not disclose ***repetitively executing a plurality of software packages at one or more rates***.

Examiner's Reply to Argument 13, Answer, p. 15.

The examiner has attempted to structure an example designed to show two successive repetition periods of a program being the same in the Blum et al. invention. and thereby demonstrate Blum et al.'s invention ***repetitively executing a plurality of software packages at one or more rates***.

The examiner starts out with four programs A, B, C, and D which are assigned time slices as follows. Each letter in the sequence identifies a time slice and the program which is scheduled to be executed during the time slice.

. . .|AAAABBCD|AAAABBCD|AAAABBCD|AAAABBCD|. . . .

Each vertical line denotes the beginning of a time-slice cycle. It is anticipated in assigning time slices that program A will require four time slices and programs B, C, and D will require the remaining four time slices in a time slice cycle, as shown below. Note that under these ideal conditions the four A's, the two B's, and each C and D have the same repetition periods and the repetition periods remain the same over time.

However, each program is allowed to take time slices from other programs. What may happen with the examiner's example over a number of time-slice cycles is shown below.

. . .|AAABBBCD|AAAABBCD|DDAABBCD|AAAABBCD|. . . .

The underlined A's denote the first A's in a four-A sequence which denotes the whole A program. In the first time-slice cycle the B program takes one of A's assigned time slices resulting in a nine time-slice period before the next execution of the A program. In the third time-slice cycle the D program takes the first two of A's assigned time slices resulting in a ten time-slice period between executions of the A program in the second and third time-slice cycles. In the fourth time-slice cycle the programs utilize their assigned time slices resulting in an eight time-slice period between executions of the A program in the third and fourth time-slice cycles.

It should be noted that the program execution period can vary from one time-slice cycle to the next. Although only one example of time slice sharing is shown above, there are many such examples that can be generated since the Blum et al. invention permits unlimited sharing of time slices between programs:

> "If it so happens that one or two of the programs are not required for a short period of time, then the computing time can be allocated among the remaining programs. Also, the individual programs can AUTOMATICALLY add to or reduce their share of the computing time. When a program adds to its computing time, the additional time is taken from one or

> more of the other programs which, at that moment, are being executed under uncritical time conditions." Blum et al., col. 5, lines 15-23, emphasis added.

Blum et al. does not disclose ***repetitively executing a plurality of software packages at one or more rates***.

Limitation [2]

Blum et al. does not disclose ***generating a sequence of time intervals for each of the plurality of software packages, the time intervals belonging to one software package not overlapping the time intervals belonging to any other of the plurality of software packages .***

Examiner's Reply to Argument 14, Answer, pp. 15-16.

The only refutation of applicants' arguments concerning limitation [2] is to make the same argument that he has made in the past—that Blum et al.'s recurring set of time slice intervals is equivalent to applicants' generating a sequence of time intervals. This assertion is simply incorrect as was discussed in the Appeal Brief:

> *"The pulse sequence generated in the Blum et al. invention (see Fig. 7, "clock pulse T") does not correspond to Limitation [2] in that Blum et al.'s pulse sequence is a single sequence that is not assigned to any program. Each of Blum et al.'s pulses merely marks the end of one time slice interval and the beginning of the next. Limitation [2] claims a plurality of sequences, each sequence being assigned to each of a plurality of software packages. In addition, each time interval in one of appellants' sequences is the time interval when the associated software package executes (see Limitation [3])." Appeal Brief, p. 38.*

The Examiner did not address applicants' invocation of 35 U.S.C. 112, ¶ 6 with respect to limitation [2]. *See In re Donaldson Co.*, 16 F.3d 1189, 29 USPQ2d 1845 (Fed. Cir. 1994)

Limitation [3]

Blum et al. does not disclose ***executing a plurality of software packages, each software package being executed during the time intervals of its sequence of time intervals.***

Examiner's Reply to Argument 15, Answer, pp. 16-17.

The examiner's comments concerning limitation [3] do not refute the arguments made in applicants' Appeal Brief.

The comments by the examiner regarding applicants' invocation of 35 U.S.C. 112, ¶ 6 with respect to limitation [3] (*see In re Donaldson Co.*, 16 F.3d 1189, 29 USPQ2d 1845 (Fed. Cir. 1994)) are supportive of the patentability of claims 1, 25, and 26:

> "There are no acts nor apparatus described in Blum et al. that are equivalent to those employed by appellants' in performing the function of Limitation [3]." Examiner's Answer, p. 16.

CLAIMS 8 AND 33

Claim 8 reads as follows:

8. The method of claim 1 wherein a software package is assigned its own dedicated memory region.

Apparatus claim 33 contains the same limitation as claim 8.

Examiner's Reply to Argument 16, Answer, p. 17.

The examiner's assertions are similar to those he has made in his office actions. Please see Appeal Brief (pp. 43-44) for a rebuttal to his assertions.

CLAIMS 23 AND 48

Claim 23 reads as follows:

23. The method of claim 1 wherein one or more of the plurality of software packages are independently compiled, linked, and loaded.

Apparatus claim 48 contains the same limitation as claim 23.

Examiner's Reply to Argument 17 Answer, p. 17.

The examiner's assertions are similar to those he has made in his office actions. Please see Appeal Brief (p. 44) for a rebuttal to his assertions.

VI. WHETHER CLAIMS 10-11, 13-14, 18, 35-36, 38-39, AND 43 ARE UNPATENTABLE UNDER 35 U.S.C. § 103(A) IN VIEW OF BLUM ET AL. (U.S. 4,109,311).

Three basic criteria must be satisfied to establish *prima facie* obviousness:

"The legal concept of *prima facie* obviousness is a procedural tool of examination which applies broadly to all arts. . . . The examiner bears the initial burden of factually supporting any *prima facie* conclusion of obviousness. If the examiner does not produce a *prima facie* case, the applicant is under no obligation to submit evidence of nonobviousness. . . . To establish a *prima facie* case of obviousness, three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art, and not based on applicant's disclosure. *In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991)." Manual of Patent Examining Procedure, 8th Edition, May 2004 Revision, § 2142.

CLAIMS 10 AND 35

Claim 10 reads as follows:

10. The method of claim 1 wherein a software package includes background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed.

Apparatus claim 35 contains the same limitation as claim 10.

Examiner's Reply to Argument 18, Answer, pp. 17-18.

The examiner's assertions are similar to those he has made in his office actions. Please see Appeal Brief (pp. 45-47) for a rebuttal to his assertions.

The examiner believes appellants' quotation from an encyclopedia in support of appellants' argument is improper. In fact, it is appellants' obligation to support their arguments with appropriate citations:

> "The appellant's contentions with respect to each ground of rejection presented and the basis for those contentions, including citations of authorities, statutes, and parts of the record relied on, should be presented in this section." *Manual of Patent Examining Procedure § 1205.02 (Appeal Brief Content).*

The appellants' have cited an article from a well-respected computer science Encyclopedia which casts doubt on the examiner's conclusion that it would have been obvious to divide the tasks performed by one of Blum et al.'s programs into foreground and background tasks and causing the background tasks to be executed after the foreground tasks have been completed. In response, the examiner seems unable to cite any authorities having contrary views.

The examiner has not established *prima facie* obviousness of claims 10 and 35.

CLAIMS 11 AND 36

Claim 11 reads as follows:

11. The method of claim 10 wherein a background task is an infinite loop.

Apparatus claim 36 contains the same limitation as claim 11.

Examiner's Reply to Argument 19, Answer, p.18.

In his previous arguments for the obviousness of claims 11 and 36, the examiner did not even assert that it is well known in the art that a background task may be an infinite loop. Now, in the

Examiner's Answer, the examiner makes this assertion even though it was pointed out in the Appeal Brief that it was not obvious that an infinite loop should be a background task:

> "On a small computer, such as is used for process control, it is common to provide a *background/foreground* system that permits two programs to execute. . . . The two programs might cooperate by mutual understanding; i.e. the programmers could insure that each would not interfere with the use of the system by the other.
>
> "While in very simple situations it may be feasible to multiprogram cooperatively, often we must be sure that, if one program somehow violates the rules of the system it does not corrupt the whole environment. Thus, most multiprogramming systems require a *monitor* (also called an *executive* or *supervisor*).
>
> **"It is the responsibility of the monitor to maintain the integrity of the system. In the case of the background/foreground system used as an example, we would like the monitor to guarantee that the foreground real-time program will be able to take its measurements, even if the background program goes into a loop and never voluntarily relinquishes control."**
>
> *Encyclopedia of Computer Science, Third Edition*, Multiprogramming, p. 908, Edited by Ralston & Reilly, IEEE Press, Van Nostrand Reinhold, New York, N.Y., 1993.

Please note the quotation above shown in boldface. Obviously, it is not a well-known fact that an infinite loop should be a background task in any computer system. Yet the examiner has not addressed any of the three criteria for establishing the *prima facie* obviousness of claims 11 and 36 other than stating that it is well known in the art, without qualification, that an infinite loop may be running as a background task.

The examiner has not established the *prima facie* obviousness of claims 11 and 36.

CLAIMS 13 AND 38

Claim 13 reads as follows:

13. The method of claim 1 wherein a failure in the execution of a software package causes information to be logged in a failure log.

Apparatus claim 38 contains the same limitation as claim 13.

Examiner's Reply to Argument 20, Answer, p.18.

The examiner argues that "it is well known in the art of program execution . . . to log execution failures linked to software packages causing the failure in order to record different occurrences, reasons, and locations of execution failures for future modifications." 10/22/04 Office Action, p. 8, § 20. However, maintaining a failure log does not appear to be an activity that is "capable of instant and unquestionable demonstration as being well-known" (*See* Manual of Patent Examining Procedure, 8th Edition, May 2004 Revision, § 2144.03 A) and thereby provide support for an obviousness conclusion. In support of this thesis appellants cited a paper by experts at Duke University which implied that maintaining a failure log was not a well-known means for achieving computer reliability and availability.

The examiner's response to appellants' Appeal Brief arguments was that appellants' citation of the Duke paper was improper. Here also, the examiner seems not to know that Appeal Brief arguments must be supported by citations to authoritative sources:

> "The appellant's contentions with respect to each ground of rejection presented and the basis for those contentions, including citations of authorities, statutes, and parts of the record relied on, should be presented in this section." *Manual of Patent Examining Procedure § 1205.02 (Appeal Brief Content).*

If the examiner believes that maintaining a failure log in computer systems like Blum et al.'s is well known, it would not be unreasonable to expect the examiner to respond with concrete evidence in support of his position, particularly in view of the Manual of Patent Examining Procedure:

> **"Ordinarily, there must be some form of evidence in the record to support an assertion of common knowledge. See *Lee*, 277 F.3d at 1344-45, 61 USPQ2d at 1434-35 (Fed. Cir. 2002; *Zurko*, 258 F.3d at 1386, 59 USPQ2d at 1697 (holding that general conclusions concerning what is 'basic knowledge' or 'common sense' to one of ordinary skill in the art without specific factual findings and some concrete evidence in the record to support these findings will not support an obviousness rejection).**

Manual of Patent Examining Procedure, 8th Edition, May 2004 Revision, § 2144.03 B.

The examiner has not established *prima facie* obviousness of claims 13 and 38.

CLAIMS 14 AND 39

Claim 14 reads as follows:

14. The method of claim 13 wherein a failure in execution is linked to the software package that caused the failure.

Apparatus claim 39 contains the same limitation as claim 14.

Examiner's Reply to Argument 21, Answer, p.18.

Appellants have been unable to find any authoritative sources for the proposition that linking a failure in execution to the software package that caused the failure in computer systems such as the one disclosed in Blum et al. is well known. Appellants cited one particular computer science encyclopedia which one would expect to have something to say on this subject and which had nothing to say.

The examiner, in his Examiner's Answer, refused to consider this information nor did he present any authoritative support for his own view that the claim limitation was well known.

The examiner took official notice that it is well known to link a failure in execution to the software package that caused the failure. 10/22/04 Office Action, p. 8, § 20. In such situations, the Manual of Patent Examining Procedure instructs:

> "If such notice is taken, the basis for such reasoning must be set forth explicitly. The examiner must provide specific factual findings predicated on sound technical and scientific reasoning to support his or her conclusion of common knowledge. *Manual of Patent Examining Procedure* § 2144.03 B.

The examiner did not set forth the basis for his reasoning in his 10/22/04 Office Action and he has not done so in the present Examiner's Answer.

The examiner has not provided "concrete evidence" to support his allegations of obviousness and the examiner's rejections of claims 14 and 39 should be reversed.

CLAIMS 18 AND 43

Claim 18 reads as follows:

18. The method of claim 1 wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software.

Apparatus claim 43 contains the same limitation as claim 18.

Examiner's Reply to Argument 22, Answer, p.18.

The examiner's assertions are similar to those he has made in his office actions. Please see Appeal Brief (pp. 52-53) for a rebuttal to his assertions.